UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/500,086	02/08/2000	Masayuki Yamasaki	43889-916	2111

20277 7590 12/18/2003

MCDERMOTT WILL & EMERY
600 13TH STREET, N.W.
WASHINGTON, DC 20005-3096

EXAMINER	
KNAPP, JUSTIN R	

ART UNIT	PAPER NUMBER
2182	

DATE MAILED: 12/18/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Applicati n No.	Applicant(s)	
	09/500,086	YAMASAKI ET AL.	
	Examiner	Art Unit	
	Justin Knapp	2182	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *10 October 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-13* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-13* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) *10*.
4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other:

DETAILED ACTION

Claim Rejections - 35 USC § 103

1. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

2. Claims 1-13 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hoge, United States Patent Number 5,930,158 in view of IBM Enterprise Systems Architecture/390 Principles of Operation.

3. Referring to claim 1, Hoge has taught:

a) a means for providing a set of instructions including an execution control instruction (column 10, lines 30-37, fig 1), the execution control instruction containing a condition field and an instruction-specifying, the instruction-specifying field defining, in binary code, the number of instructions to be executed just when the execution condition is satisfied (fig 1, element 130 and column 10, lines 46-49);

b) means for deciding whether or not the execution condition in the condition field is satisfied (column 10, lines 50-67 and column 11, lines 1-44, fig 7A); and

c) means for determining based on the outcome of the decision whether or not said number of instructions, which number has been defined by the instruction-specifying field as the number of instructions, should be nullified (column 11, lines 45-65, fig. 7A).

Hoge does not explicitly teach a condition field containing, in binary code, an execution condition. Hoge instead, teaches the condition field specifying an execution condition (fig 1,

element 140 and column 10, lines 41-44) in which the condition field is extracted from a general register. However, IBM's Enterprise Systems Architecture/390 Principles of Operation manual teaches the use of a four-bit masking field (M1, see page 7-17, Branch on Condition section) contained within the instruction code itself. The four-bit masking field is used to directly select or choose which condition code bit a pertinent instruction will test to determine if it should perform it's conditional operation. Pertinent condition codes are specified in the mask as the sum of their mask position values. For example, when the mask is 12, this specifies that a branch is made when the condition code is 0 or 1. Placement of the mask contained within the instruction code allows, as shown on page 7-17, an instruction to be conditionally dependent upon a plurality of bits of the condition code simultaneously. It would be have been obvious to one of ordinary skill in the art at the time the invention made to utilize a mask field containing execution conditions. One would have been motivated to do so because by checking for plural condition code bits simultaneously, only one instruction is used to test for multiple conditions, thereby saving program space and execution time.

4. Referring to claim 2, Hoge has taught a method comprising the steps:

a) providing an execution control instruction, the execution control instruction containing a condition field and an instruction-specifying field, the instruction-specifying field defining, in binary code, the number of instructions to be executed just when the execution condition is satisfied (fig 1, element 130 and column 10, lines 46-49);

b) deciding, based on the results of operations performed in response to one or more instructions preceding the execution control instruction currently being executed, whether or not the

execution condition in the condition field is satisfied (column 10, lines 50-67 and column 11, lines 1-44, fig 7A); and

c) determining based on the outcome of the decision step b) whether or not said number of instructions, which number has been defined by the instruction-specifying field as the number of instructions succeeding the execution control instruction, should be nullified (column 11, lines 45-65, fig. 7A).

Hoge does not explicitly teach a condition field containing, in binary code, an execution condition. Hoge instead, teaches the condition field specifying an execution condition (fig 1, element 140 and column 10, lines 41-44) in which the condition field is extracted from a general register. However, IBM's Enterprise Systems Architecture/390 Principles of Operation manual teaches the use of a four-bit masking field (M1, see page 7-17, Branch on Condition section) contained within the instruction code itself. The four-bit masking field is used to directly select or choose which condition code bit a pertinent instruction will test to determine if it should perform it's conditional operation. Pertinent condition codes are specified in the mask as the sum of their mask position values. For example, when the mask is 12, this specifies that a branch is made when the condition code is 0 or 1. Placement of the mask contained within the instruction code allows, as shown on page 7-17, an instruction to be conditionally dependent upon a plurality of bits of the condition code simultaneously. It would be have been obvious to one of ordinary skill in the art at the time the invention made to utilize a mask field containing execution conditions. One would have been motivated to do so because by checking for plural condition code bits simultaneously, only one instruction is used to test for multiple conditions, thereby saving program space and execution time.

5. Referring to claim 3, the rejections of claim 2 apply to claim 3 since claim 3 is dependent on claim 2. Furthermore, Hoge has taught the method of claim 2:

a) wherein the condition field is a single field for specifying the execution condition (fig 1, element 140 and column 10, lines 41-44), and

b) wherein the instruction-specifying field is a single field for defining the instruction number (fig 1, element 130 and column 10, lines 46-49), and

c) wherein the step c) comprises the sub-step of regarding said number of instructions as instructions to be executed just when the execution condition is satisfied, and nullifying the conditionally executable instructions if the execution condition that has been specified by the condition field is not satisfied (fig 7A, column 11, lines 27-65).

6. Referring to claim 4, the rejections of claims 2 and 3 apply to claim 4 since claim 4 is dependent on claims 2 and 3. Furthermore, Hoge has taught the method of claim 3:

a) wherein the step c) further comprises the sub-step of executing the conditionally executable instructions if the execution condition that has been specified by the condition field is satisfied (fig 7A, column 11, lines 40-44).

7. Referring to claims 5 and 6, the rejections of claim 2 apply since 5 and 6 are dependent on 2. Hoge taught a skip instruction wherein the condition field is a single field for specifying the execution condition (column 10, lines 41-44) and wherein the instruction-specifying field is a single field for defining the instruction number (column 10, lines 46-49). Hoge lacks a teaching wherein there is a first set and a second set of conditionally executable instructions wherein the first set of conditionally executable instructions is nullified and the second of set of conditionally executable instructions is executed if the execution condition is not satisfied or the first set of

conditionally executable instructions is executed and the second of set of conditionally executable instructions is nullified if the execution condition is satisfied. Hoge has taught a method wherein two skip instructions can be used to create IF/ELSEIF/ELSE logical sets of instructions (column 12, lines 16-67, figure 7C). One skip instruction is used to specify the condition and number of instructions to skip or execute for the IF instruction set. If that condition is satisfied or not satisfied, the IF instruction set is executed or nullified respectively and the ELSEIF instruction set is nullified if the IF instruction set is executed or evaluated using a second skip instruction to specify the condition and number of instructions to nullify or execute for the ELSEIF instruction set. If neither the IF or ELSEIF instruction sets are satisfied, both instruction sets are nullified and the ELSE instruction set is executed. It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the system of Hoge by removing the ELSEIF set of instructions thus eliminating the need for a second skip instruction, keeping the IF/ELSE structure to accommodate for a first and second set of conditionally executable instructions. Doing so would have allowed for use of a system incorporating Hoge's teachings using one skip instruction with a condition field and a instruction specifying field wherein there is a first set and a second set of conditionally executable instructions wherein the first set of conditionally executable instructions is nullified and the second of set of conditionally executable instructions is executed if the execution condition is not satisfied and the first set of conditionally executable instructions is executed and the second of set of conditionally executable instructions is nullified if the execution condition is satisfied. One of ordinary skill would have been motivated to do this because it eliminating a skip instruction would decrease the number of instructions to be executed thus improving the

throughput of the system rather than using two skip instructions, one for each set of conditionally executable instructions.

8. Referring to claims 7 and 8, the rejections of claim 2 apply since 7 and 8 are dependent on 2. Hoge taught a skip instruction wherein the condition field is a single field for specifying the execution condition (column 10, lines 41-44). Hoge lacks a teaching wherein the instruction-specifying field contains first and second instruction-specifying sub-fields, which respectively define first and second numbers of instructions to be executed and wherein the first number of instructions, which number has been defined by the first instruction-specifying sub-field as the number of instructions succeeding the execution control instruction, as a first set of conditionally executable instructions, and nullifying the first set of conditionally executable instructions and executing the second set of conditionally executable instructions if the execution condition specified by the condition field is not satisfied; and regarding the second number of instructions, which number has been defined by the second instruction-specifying sub-field for instructions succeeding the first set of conditionally executable instructions, as a second set of conditionally executable instructions, and nullifying the second set of conditionally executable instructions and executing the first set of conditionally executable instructions if the execution condition specified by the condition field is satisfied. Hoge has taught a method wherein two skip instructions can be used to create IF/ELSEIF/ELSE logical sets of instructions (column 12, lines 16-67, figure 7C). One skip instruction is used to specify the condition and number of instructions to skip or execute for the IF instruction set. If that condition is satisfied or not satisfied, the IF instruction set is executed or nullified respectively and the ELSEIF instruction set is nullified if the IF instruction set is executed or evaluated using a second skip instruction to specify the condition

and number of instructions to nullify or execute for the ELSEIF instruction set. If neither the IF or ELSEIF instruction sets are satisfied, both instruction sets are nullified and the ELSE instruction set is executed. It would have been obvious to one of ordinary skill in the art at the time the invention was made to eliminate the two skip instructions taught by Hoge and use one skip instruction having sub-fields in the instruction-specifying field specifying the number of instructions to execute for each instruction set in a similar system wherein there are only a first and second set of instructions to be conditionally executed. One of ordinary skill would have been motivated to do this because eliminating a skip instruction would decrease the number of instructions to be executed thus improving the throughput of the system rather than using two skip instructions, one for each set of conditionally executable instructions.

9. Referring to claim 9, the rejections of claim 2 apply to claim 9 since claim 9 is dependent on claim 2. Furthermore, Hoge has taught a method of claim 2:

a) wherein the condition field includes multiple condition sub-fields, each specifying a single associated execution condition (column 5, lines 1-20 and fig 1, element and 140 and 170 and column 10, lines 41-44), and

b) wherein the instruction-specifying field is a single field for defining the instruction number (fig 1, element 130 and column 10, lines 46-49);

c) wherein the step c) comprises a plurality of sub-steps wherein each said sub-step, said number of instructions, which number has been defined by the instruction-specifying field as the number of instructions succeeding the execution control instruction, are regarded as conditionally executable instructions, and, if the execution control condition specified by an associated one of

the condition sub-fields is not satisfied, the conditionally executable instructions at a location corresponding to the execution condition specified are nullified (column 12, lines 16-67, fig 7C).

10. Referring to claim 10, the rejections of claims 2 and 9 apply since claim 10 is dependent on claims 2 and 9. Furthermore, Hoge has taught a method of claim 9 wherein the step c) comprises a plurality of sub-steps, wherein in each said sub-step, if the execution condition specified by an associated one of the condition sub-fields is satisfied, the conditionally executable instructions at a location corresponding to the execution condition specified are executed (column 12, lines 16-67, fig 7C).

11. Referring to claims 11 and 12, the rejections of claim 2 apply since 11 and 12 are dependent on 2. Hoge has taught a method of claim 2, wherein the condition field includes multiple condition sub-fields, each specifying a single execution condition (column 5, lines 1-20 and fig 1, element and 140 and 170 and column 10, lines 41-44). Hoge lacks a teaching wherein the instruction-specifying field includes multiple instruction-specifying sub-fields corresponding to the respective condition sub-fields, each said instruction-specifying sub-field defining an associated number in binary code, and wherein the step c) comprises a plurality of sub-steps wherein each said sub-step, said number of instructions are regarded as conditionally executable instructions, and if the execution condition specified by an associated one of the condition sub-fields is not satisfied, the conditionally executable instructions at a location corresponding to the execution condition specified are nullified or if the if the execution condition specified by an associated one of the condition sub-fields is satisfied, the conditionally executable instructions at a location corresponding to the execution condition specified are satisfied. Hoge has taught a method wherein two skip instructions can be used to create IF/ELSEIF/ELSE logical sets of

instructions (column 12, lines 16-67, figure 7C). One skip instruction is used to specify the condition and number of instructions to skip or execute for the IF instruction set. If that condition is satisfied or not satisfied, the IF instruction set is executed or nullified respectively and the ELSEIF instruction set is nullified if the IF instruction set is executed or evaluated using a second skip instruction to specify the condition and number of instructions to nullify or execute for the ELSEIF instruction set. If neither the IF or ELSEIF instruction sets are satisfied, both instruction sets are nullified and the ELSE instruction set is executed. It would have been obvious to one of ordinary skill in the art at the time the invention was made to have modified the teachings of Hoge. Doing so one of ordinary skill would eliminate the need for multiple skip instructions each having a different instruction-specifying field to be associated with a corresponding condition field and use one skip instruction having multiple sub-fields in the instruction-specifying field specifying the number of instructions to be conditionally executed based on an associated condition specified by multiple sub-fields in the condition field. One of ordinary skill would have been motivated to do this because eliminating a skip instruction would decrease the number of instructions to be executed thus improving the throughput of the system rather than using multiple skip instructions.

12. Referring to claim 13, Hoge has taught a method comprising:

a) providing a set of instructions including an execution control instruction, the execution control instruction containing a condition field with an execution condition in binary code (as taught herein above, see rejection of claim 3);

b) deciding, based on the results of operations performed in response to one or more instructions preceding the execution control instruction in the instruction set provided, whether or not the

execution condition in the condition field of the execution control instruction is satisfied (as taught herein above, see rejection of claim 3); and

c) nullifying one or more instructions succeeding the execution control instruction if the execution condition that has been specified by the condition field of the execution control instruction is not satisfied (as taught herein above, see rejection of claim 3).

Hoge does not explicitly teach a condition field containing, in binary code, an execution condition. Hoge instead, teaches the condition field specifying an execution condition (fig 1, element 140 and column 10, lines 41-44) in which the condition field is extracted from a general register. However, IBM's Enterprise Systems Architecture/390 Principles of Operation manual teaches the use of a four-bit masking field (M1, see page 7-17, Branch on Condition section) contained within the instruction code itself. The four-bit masking field is used to directly select or choose which condition code bit a pertinent instruction will test to determine if it should perform it's conditional operation. Pertinent condition codes are specified in the mask as the sum of their mask position values. For example, when the mask is 12, this specifies that a branch is made when the condition code is 0 or 1. Placement of the mask contained within the instruction code allows, as shown on page 7-17, an instruction to be conditionally dependent upon a plurality of bits of the condition code simultaneously. It would be have been obvious to one of ordinary skill in the art at the time the invention made to utilize a mask field containing execution conditions. One would have been motivated to do so because by checking for plural condition code bits simultaneously, only one instruction is used to test for multiple conditions, thereby saving program space and execution time.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Justin Knapp whose telephone number is (703) 308-6132. The examiner can normally be reached on Mon - Fri 9 am - 5:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Jeffrey Gaffin can be reached on (703) 308-3301. The fax phone number for the organization where this application or proceeding is assigned is (703) 746-7239.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

Justin Knapp
Examiner
Art Unit 2182

December 11, 2003

JEFFREY GAFFIN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100